1. (8 points) Basic concepts.
   A. It is better to assume services and business partners are static so as to promote flexibility in an IT system
      Solution: False:
   B. Locality is the idea of reducing dependencies across services to the extent possible while ensuring coherence
      Solution: True:
   C. Reputation is an important consideration for selecting a service provider but not for selecting a service requestor
      Solution: False:
   D. Context only matters when there is high potential cost to the user of a wrong decision by a mobile application
      Solution: False:

2. (18 points) Agents.
   A. Autonomy goes hand in hand with accountability: a party that is autonomous is the one that can be held accountable for its actions
      Solution: True:
   B. The persistence of an agent’s identity is crucial to its ability to participate in service engagements
      Solution: True: We talked about it in terms of long-lived conversations, which are necessary to realize service engagements
   C. The decomposition of a goal relies upon nothing more than the AND and OR operators explicitly stated within the description of a goal
      Solution: False: Such operators are rarely explicit and goal decomposition in general involves domain knowledge
   D. When we apply cognitive agents to services, we can naturally map an agent’s sensors to services
      Solution: True:
   E. The IOPE (or IOPR) representation of services does not support planning to compose services
      Solution: False:
   F. Many of control constructs defined in the OWL-S vocabulary encode a scripting language
      Solution: True:
G. An inference rule provides a simple means of progressing from one state to the next

Solution: False: inference rules do not bring about state transitions

H. Backward chaining is a control regime for rule-based reasoning that produces bindings, if any, that make its consequent true

Solution: True:

I. Encoding business policies using rules provides a simple means to inspect policies and verify whether an agent (implementation) is applying the appropriate policies

Solution: True:

3. (18 points) Multiagent systems and organizations.

A. The main shortcoming of the mentalist approach (as in FIPA) is that it confuses architecture with implementation

Solution: True:

B. A suitable notion of consistency for a multiagent system is that each of the agents in a system be locally consistent

Solution: False: should be local and shared consistency

C. When a datum is marked EXTERNAL that signifies another agent is the authority regarding whether that datum is IN or OUT

Solution: True: should be local and shared consistency

D. Organizations provide a basis for establishing consistency of service interactions across multiple agents

Solution: False: just coherence of computations of service interactions

E. Modeling virtual enterprises as organizations provides a basis for recovering from business exceptions

Solution: True:

F. A (communicative) convention maps physical actions to social actions

Solution: True:

G. A mentalist semantics for communications is ideally suited to open systems, such as in service-oriented computing

Solution: False:

H. Commitments and operations on them can provide a standard of correctness for the messages in a business protocol such as bid or accept
Solution: True:

I. A problem with assigning social meaning to the messages in a business protocol is that it is not possible to determine whether the agents participating in a protocol comply with the social meaning

Solution: False:

4. (20 points) Protocols.
   
   A. The good feature of standard messaging middleware for services is that they make sure two services will always observe any common messages in the same order

   Solution: False: messaging middleware allows for the possibility that messages may cross

   B. An idea behind BSPL is that all the shared information an agent needs to play a role in a protocol is passed explicitly through message parameters

   Solution: True:

   C. A simple protocol such as Hello does not need a key

   Solution: False:

   D. In LoST, an agent playing a role may have “too much” information to be able to send a message

   Solution: True:

   E. To construct protocols for cases such as negotiation (e.g., involving offer and counteroffer messages about price), we must allow certain parameters (e.g., price) to be overwritten for the same key

   Solution: False:

   F. In our architectural assumptions underlying LoST, an agent who already knows the binding of a parameter is forbidden from receiving a message with an \[\text{out}\] binding of the same parameter

   Solution: False:

   G. In LoST, an agent can gain or lose knowledge about parameter bindings as it sends and receives messages

   Solution: False: an agent can never lose knowledge of parameter bindings

   H. A protocol that is safe is guaranteed to complete in all enactments

   Solution: False:

   I. A protocol that is live is guaranteed to complete in all enactments
Solution: False: only if the agents enacting the roles decide to send some enabled messages and no such messages are lost

J. The only way an agent learns the binding of a parameter is if it sends or receives a message with that parameter

Solution: True:

5. (30 points) Consider the following protocol for health care services.

\[
\text{Health Insurance} \quad \{ \quad \text{role P, H, I} \quad // \text{Patient, Health care provider, Insurance company} \\
\text{parameter} \quad ... \quad \text{Part (a)} \quad ... \\
\]

\[
P \rightarrow I: \text{apply} [\text{in patientID}] \\
I \rightarrow P: \text{policy} [\text{in patientID}, \text{out policyID}] \\
P \rightarrow H: \text{visit} [\text{in patientID}, \text{in policyID}, \text{out problem}] \\
H \rightarrow P: \text{plan} [\text{in patientID}, \text{in policyID}, \text{in problem}, \text{out procedureID}] \\
H \rightarrow I: \text{verify} [\text{in patientID}, \text{in policyID}, \text{in procedureID}] \\
I \rightarrow H: \text{approve} [\text{in policyID}, \text{in procedureID}, \text{out approvalID}] \\
H \rightarrow P: \text{procedure} [\text{in patientID}, \text{in policyID}, \text{in problem}, \text{in procedureID}, \text{in approvalID}, \text{out done}] \\
\}
\]

Treat the problem parts below as independent of each other.

(a) (10 points) The above protocol is designed to be composable so that all its parameters are public. State which is "in" and which is "out" so as to complete the parameter line.

Solution: Deduct 2 points for each parameter omitted or wrongly adorned up to a maximum deduction of 10 points.

parameter in patientID, out policyID, out problem, out procedureID, out approvalID, out done

(b) (10 points) Modify the above protocol so that the INSURANCE COMPANY may deny a request from a health provider.

Solution: Deduct 2 points for role omitted and for each parameter omitted or wrongly adorned, and for each superfluous parameter—up to a maximum deduction of 10 points.

Insert a new message

\[
I \rightarrow H: \text{deny} [\text{in policyID}, \text{in procedureID}, \text{out approvalID}, \text{out done}] \\
\]

(c) (10 points) Modify the above protocol so that the PATIENT may specify a treatment plan, i.e., a desired procedure to be performed by the HEALTH CARE PROVIDER.
Solution: Deduct 2 points for role omitted and for each parameter omitted or wrongly adorned, and for each superfluous parameter—up to a maximum deduction of 10 points.

Create two variants of the visit message

P → H: visit [in patientID, in policyID, out problem, nil procedureID]
P → H: visit [in patientID, in policyID, out problem, out procedureID]